OptiqueVQS – Towards an Ontology-based Visual Query System for Big Data

Ahmet Soylu*
University of Oslo
Oslo, Noway
ahmets@ifi.uio.no

Martin Giese
University of Oslo
Oslo, Noway
martingi@ifi.uio.no

Ernesto Jimenez-Ruiz
University of Oxford
Oxford, UK
ernesto@cs.ox.ac.uk

Evgeny Kharlamov
University of Oxford
Oxford, UK
fname.lname@cs.ox.ac.uk

Dmitry Zheleznyakov
University of Oxford
Oxford, UK
fname.lname@cs.ox.ac.uk

Ian Horrocks
University of Oxford
Oxford, UK
fname.lname@cs.ox.ac.uk

ABSTRACT

A recent EU project, named Optique, with a strong industrial perspective, strives to enable scalable end-user access to Big Data. To this end, Optique employs an ontology-based approach, along with other techniques such as query optimisation and parallelisation, for scalable query formulation and evaluation. In this paper, we specifically focus on end-user visual query formulation, demonstrate our preliminary ontology-based visual query system (i.e., interface), and discuss initial insights for alleviating the affects of Big Data.

Categories and Subject Descriptors

H5.2 [[**Information Interfaces and Presentation**]: User Interfaces; H5.4 [[**Information Interfaces and Presentation**]: Hypertext/Hypermedia

General Terms

Design, Human Factors

Keywords

Visual Query Systems, Ontologies, Big Data

1. INTRODUCTION

In an enterprise context, engaging employees directly with data could substantially increase competitiveness and profitability by augmenting the *value creation* potential (cf. [27]). However, *data access* is still a major bottleneck for many organisations. This due to the sharp distinction between employees dedicated to extracting data (i.e., *database/IT experts, skilled users* etc.), and those dedicated to interpreting and using data (i.e., *domain experts, end-users* etc.). Since domain experts mostly do not posses necessary competences to formulate queries by using structured query languages such as *SQL*, the following scenarios are commonly practiced (see Figure 1). The first scenario (i.e., simple case) involves uniform data sources and predefined queries embedded into applications; while the second scenario (i.e., complex case) follows a *man-in-the-middle* approach; domain experts communicate their *information needs* to IT experts, who in turn translate them to formal queries over disparate data sources. The former is limited to the enumeration of possible information needs, while the latter results in lengthy turn around times in the range of days to weeks (cf. [10]). Therefore, it is important to support domain experts with intuitive and natural visual data access tools for extracting data from legacy data sources.

Visual query formulation (cf. [8]) is built on the idea of the *direct manipulation* (cf. [32]) of visual objects representing the domain elements and is long studied in the literature. However, early approaches mostly suffer from the abstraction levels they operate on; database schemas, object-oriented models etc. are not meant to capture a domain per se and are not truly natural for end-users. The use of ontologies as a natural communication medium for end-users emerged as a prominent approach; however, early attempts (e.g., [9, 1]) remained at experimental stage and did suffer from the lack of appropriate frameworks for bridging ontologies and relational data sources. The picture is almost complete with the *Semantic Web* and *ontology-based data access* (OBDA) technologies (cf. [29, 22]) that bridge ontologies and relational data sources. Ontology-based visual query formulation and data access (e.g., [9, 26]) is still an active research domain; however, the challenge is further exacerbated by the *Big Data effect* (cf. [24]), which is characterised through the *volume*, *complexity*, *variety*, and *velocity* dimensions of data as well as its schemata. The Big Data not only introduces efficiency problems for query evaluation, but also perceptual and cognitive problems for visual query formulation; it becomes harder to communicate and represent large, complex, and varied domain knowledge and data to the end-users.

A recent EU project, named *Optique – Scalable End-user*

*Corresponding author.

MEDES'13 October 28-31, 2013, Neumünster Abbey, Luxembourg

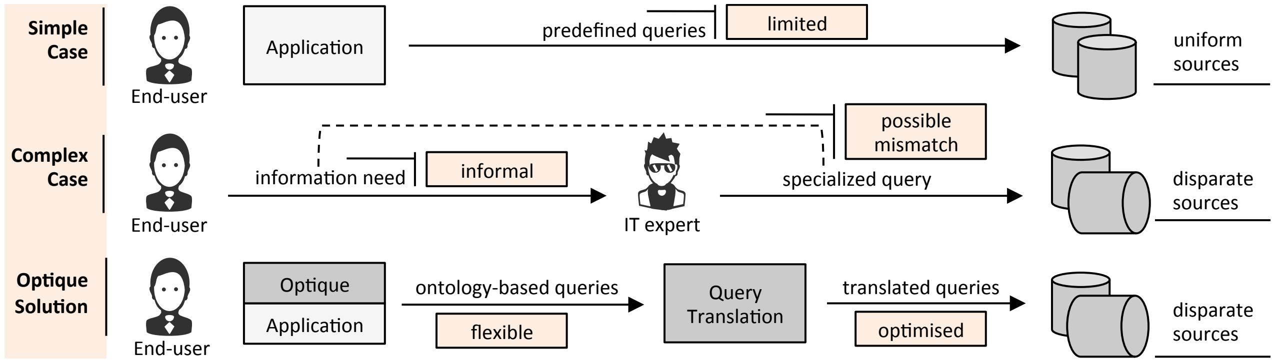

Figure 1: Common data access scenarios in enterprise context and the Optique approach.

Access to Big Data[1], strives to alleviate the aforementioned challenges and employs an ontology-based approach, along with other techniques such as query optimisation and parallelisation, for scalable query formulation and evaluation [12]. The Optique approach for ontology-based visual query formulation and data access is depicted in Figure 1 (i.e., Optique solution) and Figure 2. Users interface with the system through a *visual query system* (VQS), which is a system of interactions, rather than a formal language (i.e., *visual query language* – VQL), that generates the underlying textual language – e.g., SPARQL (cf. [11]). The OptiqueVQS relies on an OBDA framework, which is not in the scope of this paper (cf. [12]), that allow access to relational data over ontologies. However, briefly, as depicted in Figure 2, once translated into a linguistic structure a query is passed through two *rewrite* phases to transform it into a *complete*, *correct*, and *highly optimised* query over data sources (cf. [29, 28]). The first phase rewrites the query by taking ontology constraints into account, while the second one translates the query into the language of underlying data sources (e.g., SQL) through *mappings* defined between the ontology and data sources.

The project has two industrial partners, namely *Statoil*[2] and *Siemens*[3], which provide real-life use cases from energy domain. In this paper, we specifically focus on end-user visual query formulation, demonstrate our preliminary ontology-based visual query system *OptiqueVQS* (i.e., interface) over a Statoil use case, and discuss initial insights for alleviating the affects of Big Data. We use the semantic representation of *Norwegian Petroleum Directorate*'s database[4], that contains public data concerning the petroleum activities on the Norwegian continental shelf (cf. [33]).

The rest of the paper is structured as follows. Section 2 provides an elaborate view on challenges and requirements for visual query formulation both from a generic and Big Data perspective. Section 3 presents the Optique approach and the proposed visual query system, while Section 4 introduces the design rationale behind OptiqueVQS. Section 5 provides a discussion on OptiqueVQS from expressiveness and usability point of view and enumerates possible approaches to address the Big Data issues. Finally Section 6 concludes the paper and provides pointers to other components that form the OBDA framework of Optique.

[1]http://www.optique-project.eu
[2]http://www.statoil.com
[3]http://www.siemens.com
[4]http://sws.ifi.uio.no/project/npd-v2/

2. CHALLENGES AND REQUIREMENTS

For a visual query system, *expressiveness* and *usability* form an inclusive frame that spans the main challenges and requirements (cf. [8]). Expressiveness defines the ability and breadth of a language or system to characterise the domain knowledge and information need, while usability defines capability of a system to meet its identified aim with *effectiveness* (i.e., doing the right things), *efficiency* (i.e., doing the things right), and *user-satisfaction* (i.e., the perceived quality of the interface and dialog). There are two type of data access activities in a visual query formulation scenario, which should be elaborated from expressiveness and usability perspectives. These are *exploration*, which relates to the activities for understanding and finding schema concepts and relationships relevant to information need; and, *construction* (i.e., formulation), which concerns the compilation of relevant concepts and constraints into formal information needs (cf. [7]).

Concerning expressiveness, for a system or language, a piece of domain knowledge or query type makes value, only if end-users need, understand, and use it. Consequently, it is appropriate to approach expressiveness from a user perspective. Therefore, we are primarily interested in the types of requests and knowledge that a specific visual query system or language should accommodate (i.e., necessity/need and practicality/complexity), rather than the functional capability of underlying formality and its analysis and development (i.e., ability). Regarding expressiveness in exploration activities in an ontology-based setting, the main question concerns the selection of ontology constructs and semantics that should be communicated at the interface level. Some constructs and semantics are meaningful for aiding users during query construction (e.g., disjointness axioms might useful to communicate to end-users, since they prevent users to select two disjoint classes for the interpretation of same query variable), while on the other hand others might be only meaningful at the query answering stage (e.g., transitivity axioms might not be valuable to communicate, since end-users are only interested in the intended meaning rather than its realisation). Another concern in this line is the *propagative effect* of semantics (cf. [13]). In an ontology, explicit restrictions attached to a concept will be inherited by its subconcepts (i.e., *top-down propagation of property restrictions*), and the interpretation of a concept also includes the interpretations of all its subconcepts (i.e., *bottom-up propagation of property restrictions*). Therefore, for a given concept, it

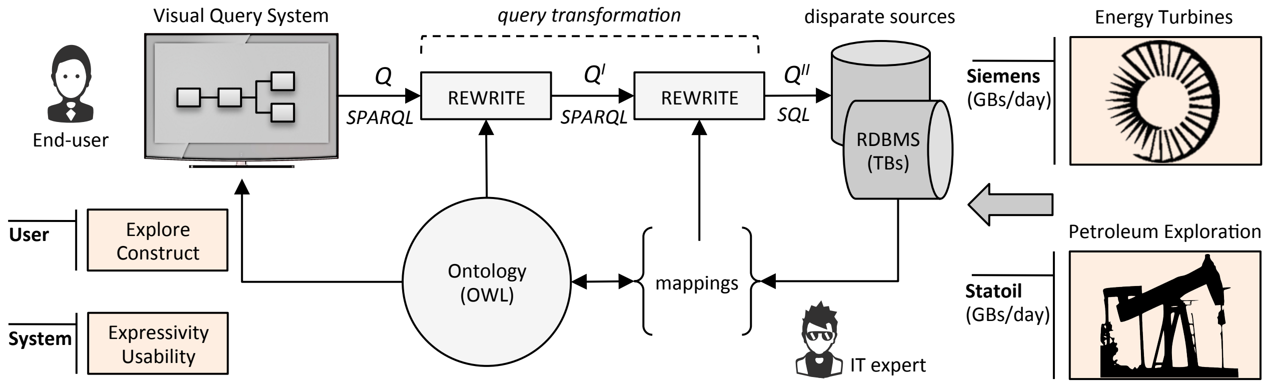

Figure 2: Ontology-based visual query formulation and ontology-based data access.

may also make sense to suggest the (potential) restrictions of its subconcepts and superconcepts. However, it becomes challenging, since the amount of information to communicate increases drastically. Regarding expressiveness in construction activities, queries can be considered in different topological forms, such as *linear queries* and *queries with branching*. The latter includes *conjunctive queries* and *disjunctive queries* that include path expressions merged with *AND connective* and *OR connective* respectively. Another notable form is *cyclic queries*, in which at least two existentially quantified independent variables range on the interpretation of the same concept-node in the ontology graph (cf. [8]). There are also query types for which there is no simple way to formulate them purely with the aforementioned topological forms, such as queries with *quantification* (i.e., *universal* and *existential*), *negation*, and *aggregation* (e.g., *count*, *sum*, *max* etc.). The overall challenge here is to find an appropriate compromise between the need and complexity, that is only supporting the classes of queries, which are reasonably complex for end-users and do address substantial amount of information needs.

Concerning usability, the challenge is to select and intertwine representation metaphors, visual attributes, and interaction styles that require less knowledge, skills and learning effort, and allow users to *discern*, *comprehend*, and *communicate* the maximum amount of information effortlessly. Ontology visualisation techniques and approaches play a crucial role in this context (cf. [17]). However, the Big Data effect, which not only concerns data but also its schemata, impedes the use of visual query systems and languages. Primarily, the volume and complexity of domain knowledge (e.g., schema, ontology etc.) hinders human perception and cognition respectively. Secondly, the variety raises the need for more domain-specific presentations and interaction experiences adapted to data at hand at any moment, while the velocity dimension requires data access systems to address reactive scenarios, where data is automatically *detected*, *assessed* and *acted upon*. Such reactive scenarios likely to involve queries that do not have any matching results at the time of authoring but are supposed to detect future possible occurrences (e.g., data that describes a fault). A data access system should provide a wide range of support for situating and orienting users in the conceptual space in order to help them to understand and make use of data, and it should be integrated and adapted to *context*, such as *personal*, *data-related*, *task-related*, *organisational*, and *environmental*. Regarding the usability dimension in exploration activities, the volume and complexity mostly matter in terms of schemata rather than the data, since users interact with the system primarily at a conceptual level. Exploration with a large number of concepts, relationships, and attributes with high complexity is a hard problem, since the presentation could easily become *overcrowded* and *cluttered* and prevent user to reach an overall understanding. The high variety and velocity have implications on data access systems at exploration stage. Specific representations that are best suited to the nature of data at hand, along with generic presentation facilities, are required to better communicate and interact with different types of data. It is also important to allow end-users to explore a domain at instance level in order to help them to gain insights on the underlying data. Regarding the usability of construction, the challenge is to guide users to their targets with minimum amount of deviations and backtracks. The largeness of domain knowledge together with top-down and bottom-up propagation of property restrictions, that we have discussed previously, increase the number of possibilities enormously. At any step of query construction, the users are confronted with high number of concepts and properties to choose from. This reduces the ability of users to quickly decide on the next step. Another challenge is to drive *user attention*, which is a precious resource, within the large streams of data (cf. [27]), so that valuable data fragments could be exploited. Finally, the usability of an ontology, which steers the interface, is an important consideration. Yet, the usability aspects of ontologies remain unnoticed to a large extent; the mismatch/gap between users' understanding of domain and an ontology could easily hamper the success of a well-designed interface.

Overall, from the expressiveness perspective, one should realise that a visual query system or language, meant for naive end-users, should primarily match the level of users and therefore is likely to be less expressive than the underlying formal linguistic language (e.g., SPARQL). From usability perspective, a visual query system should drive the capabilities of the output medium and human visual system at an optimum level, while bridging the gap between the domain representation and user mental model.

3. OPTIQUE APPROACH

OptiqueVQS is designed as a *user-interface (UI) mashup* built on *widgets*. A UI mashup aggregates different applica-

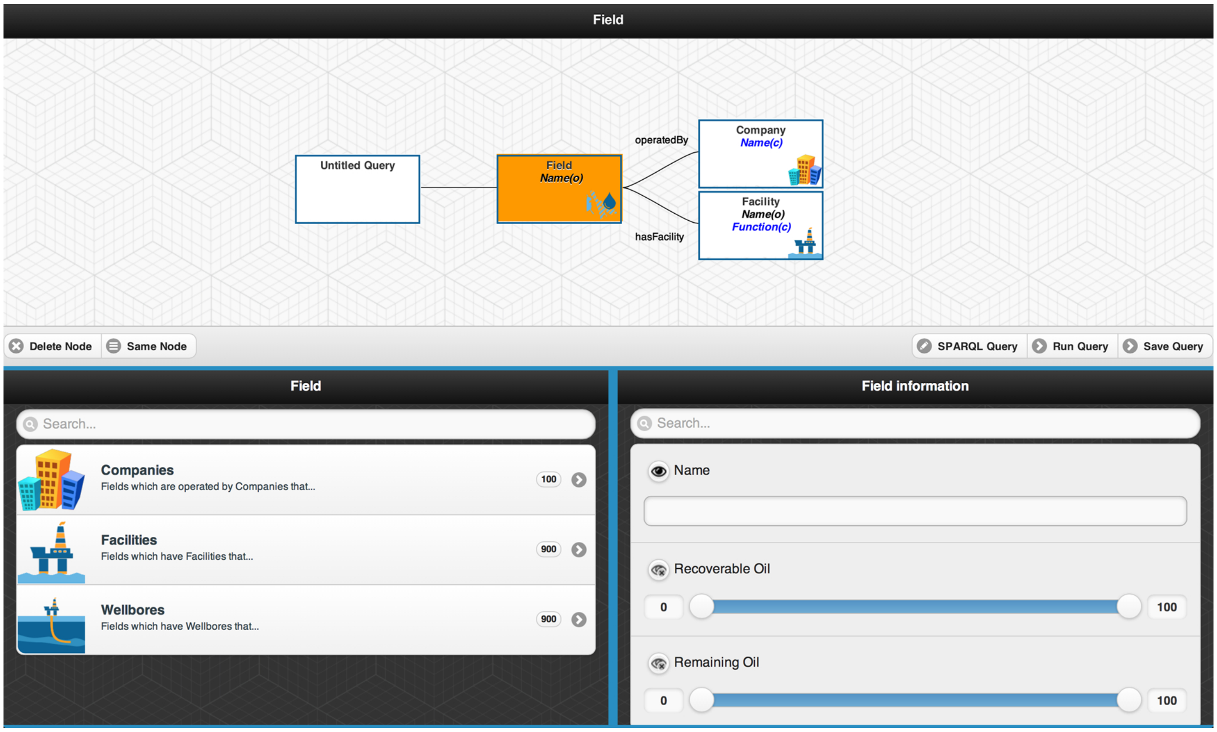

Figure 3: OptiqueVQS – an example query is depicted for the Statoil use case.

tions into a common graphical space and *orchestrates* them for common goals (cf. [34]). Widgets[5] are the building blocks of our VQS and refer to *portable*, *self-contained*, *full-fledged*, and mostly client side applications with limited functionality and complexity. Widgets in our system communicate with each other by delivering events, generated by user actions, through a client-side *communication channel*. Each widget reacts to events either in a preprogrammed way or by considering the *semantic and syntactic signatures* of events.

We initially have three widgets in our system as depicted in Figure 3. The first widget (W1 - see the bottom-left part of Figure 3) is a *menu-based query by navigation* widget and allows users to navigate concepts through pursuing relationships between them, hence joining relations in a database. The second widget (W2 - see the bottom-right part of Figure 3) is a *form-based* widget, which presents the attributes of a selected concept for *selection* and *projection* operations. The third widget (W3 - see the top part of Figure 3) is a *diagram-based* widget and provides an overview of the constructed query and affordances for manipulation. These three widgets are orchestrated by the system, through harvesting event notifications generated by each widget as a user interacts, to jointly extract and represent the information need of a user.

In a typical query construction scenario, a user first selects a *kernel concept*, i.e., the starting concept, from W1, which initially lists all domain concepts accompanied with icons, descriptions, and the potential/approximate number of results. The selected concept becomes the *focus/pivot* concept (i.e., the node coloured in orange or highlighted), appears on the graph (i.e., W3) as a *variable-node*, W2 displays its attributes, and W1 displays all concept-relationship pairs pertaining to this concept. The user can select attributes to be included in the result list (i.e., using the "eye" button) and/or impose constraints on them through form elements (i.e., W2). Currently, the attributes selected for output appear on the corresponding variable-node in black with a letter "o", while constrained attributes appear in blue with letter "c". Note that W1 does not purely present relationships, but combine relationship and concept pairs (i.e., relationship and range) into one selection; this helps us to reduce the number of navigational levels that a user has to pass through. The user can select any available option from the list, which results in a join between two variable-nodes over the specified relationship and moves focus to the selected concept (i.e., pivot). The user has to follow the same steps to involve new concepts in the query and can always jump to a specific part of the query by clicking on the corresponding variable-node. The arcs that connect variable-nodes do not have any direction, since for each active node only outgoing relationships, including inverse relationships, are presented for selection in W1; this allows queries to be always read from left to right. In W3, we employ node duplication approach for cyclic queries for the sake of having tree-shaped representations for queries, hence avoiding graph representation, which might be complex for end-users to comprehend. An example query is depicted in Figure 3 for the Statoil use case. The query asks for all fields that contain an oil producing facility and are operated by the Statoil company. In the output, we would like to see the name of the field and the name of the facility.

The user can delete nodes by switching to delete mode or assert that two variable-nodes indeed refer to same variable (i.e., cyclic query). Affordances for these are provided by the buttons at the bottom-left part of the W3. The user

[5]http://www.w3.org/TR/widgets/

can also switch to SPARQL mode and see the textual from of the query by clicking on "SPARQL Query" button at the bottom-right part of the W3 as depicted in Figure 4. The user can keep interacting with the system in textual form and continue to formulation process by interacting with the widgets. For this purpose, pivot/focus node is highlighted and every variable-node is made clickable to allow users to change focus. Currently, the textual SPARQL query is non-editable and is for didactical purposes, so that advanced end-users, who are eager to learn the textual query language, could switch between two modes and see the new query fragments added after each interaction. There are also plans for enabling users to edit the query text in SPARQL mode, which is discussed in Section 5.

An extensive demo of the previous prototype (the SPARQL view is not included) is available online[6] as a video.

4. DESIGN RATIONALE

The current design of the interface and architecture is meant to address core requirements and to provide a good basis for the accommodation of others. As stated earlier, we intentionally opt for a visual query system rather than a formal visual query language, since a system-based approach could provide more possibilities in terms of usability and expressivity than a language-based approach. This is because a systemic approach allows us to avoid rigid boundaries of a formal language and places less burden on users by relying on their general semantic knowledge (i.e., on manipulating objects and computer interaction) rather than their ability to learn new language and syntax (cf. [32]).

The *visual representation* and *interaction paradigms* (cf. [8]), along with underlying *metaphors*, *analogies* etc., are of primary importance for a VQS. We have observed that a single representation and interaction paradigm is not sufficient for addressing main data access activities, i.e., exploration and construction, at an acceptable level of *expressiveness* and *usability*. Therefore, we strive to combine the best parts of different paradigms for developing a successful query formulation interface. The architectural choice of Optique plays a crucial role in this respect, a mashup approach built on widgets is meant to ensure *flexibility* and *extensibility*, so that we can combine different representation and interaction styles. The core benefits of such an approach are that it becomes easier to deal with the complexity, since the management of functionality and data could be delegated to different widgets; each widget could employ a different representation paradigm that best suits its functionality; widgets could be used alone or together, in different combinations, for different contexts and experiences; and the functionality of the overall interface could be extended by introducing new widgets.

Although a limited amount work exists on ontology-based visual query formulation, the large amount of work on linked data browsing and search, available in the Semantic Web domain, provides considerable feedback for the design choices underlying OptiqueVQS (cf. [3]). *Faceted search* (cf. [37]) and *Query by Navigation* (QbN) (cf. [36]) are prominent approaches in this respect. Faceted search, being an advanced *form-based* approach, is based on series of orthogonal dimensions that can be applied in combination to filter the information space; each dimension, called *facet*, corresponds to a taxonomy. In its most common form, each facet option is accompanied with the number of accessible instances upon a possible selection. This is to prevent users from reaching empty result sets. QbN exploits the graph-based organisation of information to allow users to construct queries by traversing the relationships between concepts. Each navigation from one concept to another is indeed a join operation. Actually, end-users are quite familiar with both types of search approaches; faceted search is widely used in commercial websites such as eBay and Amazon for listing and filtering products, while the navigation is the backbone of web browsing. *Tabulator* [2] and *SEWASIE project* [9], and *Flamenco* [39] and *mSpace* [31] are well-known examples of QbN and faceted search respectively. The examples of QbN provide weak or no support for select and projection operations; similarly the examples of faceted search do not provide sufficient support for joining concepts. Hybrids of both are also available and combine the power of both paradigms, such as *VisiNav* [15] and *OZONE* [35]. Nevertheless, these approaches, due to their nature, are highly explorative and instance oriented. That is, firstly, the navigation is mostly for data browsing purposes; a final query, which encompasses the visited concepts, is not generated. Hence, there is no clear distinction between explorative and constructive user actions and there is a lack of support for *view* (i.e., the active phase of a query task) and *overview* (i.e., the general snapshot of a query task). Secondly, frequent interaction with the data is required (i.e., database-intensive), which is problematic with large scale data sources.

Although aforementioned works do not address exactly same research challenge and the domain, they already offer a lot in terms of their success with end-users and form a basis for comparison. The interface described in this paper adapts and improves basic and core techniques from these works. Concerning the interface design, from a holistic perspective, OptiqueVQS aims to provide a clear distinction and support for view and overview. In this context, W3 constantly provides an overview of the query, while W1 and W2, always being focused on a certain concept, enable users to iteratively formulate their queries. Concerning widgets on an individual basis, representation and interaction styles are of crucial importance (cf. [8, 17]). W1 follows a *list/menu-based* representation style and enables us to present considerably higher number of items to the users in an effective and efficient manner. End-users are expected to be familiar with the navigational interaction style employed by W1 as stated earlier (cf. [36]). The form-based representation style and range selection interaction style employed by W2 are well-known by the end-users and known to be intuitive (cf. [8]). We provide only a limited amount of faceted search flavour, since frequent database access is not feasible in our context. Finally, the diagram-based approach employed by W3 is good at communicating relationships over a spatial dimension and provides an intuitive overview (cf. [17]).

5. DISCUSSION AND OUTLOOK

An ontology-based visual query system falls into category of *ontology-driven information systems* (cf. [30]) and visual query formulation is an *end-user development/programming practice* (cf. [23]). The challenge is one of usability and we believe that ontologies have a lot to offer in this respect. Although OptiqueVQS relies on an OBDA framework, by principle it is suitable for any graph-based data

[6] http://sws.ifi.uio.no/project/optique/pubshare/medes2013/

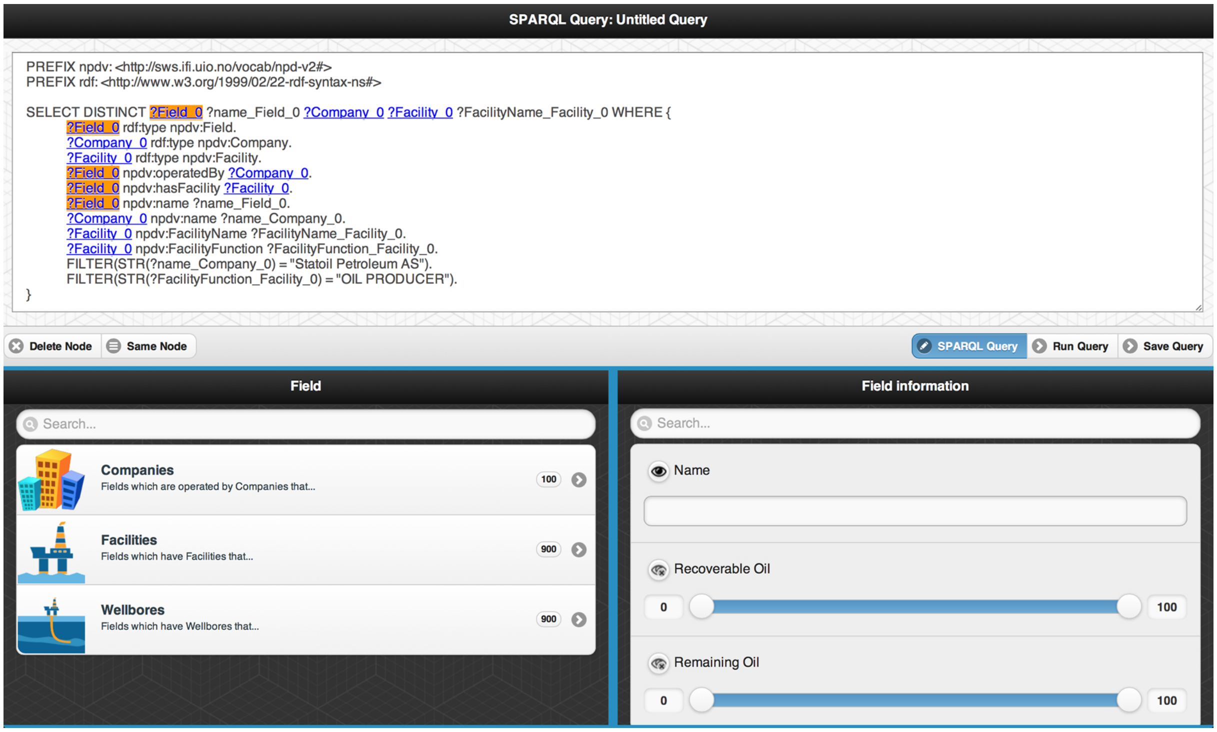

Figure 4: OptiqueVQS – the example query is depicted in SPARQL form.

model. However, an ontology-based approach is expected to provide maximum benefit due the power of ontologies in knowledge representation and reasoning, which provide a natural end-to-end semantic connection between end-users and data sources and ability to relate the whole set of implied information instead of what is explicitly stated and available in data sources. The development and improvement of OptiqueVQS is still ongoing, and in what follows we discuss expressivity and usability issues to be addressed.

Regarding expressiveness, we categorise queries into three levels with respect to the perceived complexity and need. This categorisation is based on our personal experience and should be reorganised or verified experimentally. First level corresponds to simple *three-shaped* conjunctive queries, while the second level refers to cyclic queries, disjunctive queries, and aggregation. The last level corresponds to queries with universal quantifiers, and negation. We postulate that most of the end-user queries will be centred around first level. The current interface addresses first level queries and partially second level queries with the inclusion of cycles. However, we do see possibilities to address second level and third level queries to support advanced users. For this purpose, we pursue a *layered/spiral* approach, where the advanced functionalities of system are organised into a set of layers (cf. [32]). A user could start using a system with minimal functionality and unlock new functionalities as his/her competence progresses. Another approach is to employ *collaborative query formulation/search* and *query reuse* (cf. [41, 20]) to enable domain experts and IT experts to formulate queries collaboratively and to reuse existing queries formulated by their peers. IT experts are likely to use a textual query editor, while domain experts are expected to use the visual query system, where the challenge is to ensure synchronisation between these two modes (cf. SPARQL mode).

Regarding usability, particularly for large ontologies, guiding users among hundreds of concepts, attributes, and relationships is of crucial importance. At this stage, the current proposal attacks the query formulation challenge itself; the work for addressing the Big Data effect is under progress. One apparent approach, which we follow, is to enable users to interact with the system in a way that they gradually access and explore the ontology and data – e.g., by expanding and retracting nodes. On top of that, *adaptation* and *recommendation* techniques (cf. [4]) could give us chance to prune the conceptual space for the user, for which we could harvest query logs, data, and a set of heuristics to rank and filter out irrelevant concepts, relationships etc. Another approach would be the use of *schema clustering* and *summarisation* techniques (cf. [6, 40]). The former aims at automatically adding abstraction layers to conceptual schemas, after which users are not confronted with hundreds or thousands of concepts, but with high level clusters that they could drill down, while the latter is meant to provide a visual overview of the entire domain to aid user understanding. Finally, domain-specific presentations both at data and conceptual level are expected to facilitate communication, for which the proposed widget-based architecture opens up the possibility.

6. CONCLUSION AND FUTURE WORK

We have presented a preliminary ontology-based VQS. The *multi-paradigm* approach built on UI mashups allows us to provide a good balance between view and overview. Moreover, each representation paradigm could handle different kinds of ontology constructs, for which it is best suited to. We follow an iterative user-centric development approach; we have already had first informal experiences with the domain experts (i.e., end-users) of Statoil. Early impressions suggest that the overall approach is promising and end-users are able to use OptiqueVQS. Some minor improvements are suggested not only at interface level, but also at the ontological level, which is inline with our claim on the usability of ontologies. Formal experiments are planned to be executed on non-employee users first (e.g., students) and later on employees after the first cycle of revisions, since the time of domain experts are scarce and valuable.

Future work not only includes improvements concerning expressiveness and Big data affect, but also the development of components (i.e., widgets) for exploring domain both at a conceptual and data level (cf. [38, 25]). Currently, OptiqueVQS does not provide any explicit support for the exploration of ontology and underlying data. Exploration support is planned to be in two categories: the first category refers to interactive visualisation support, scalable to large ontologies, that allows users to gradually explore the ontology and underlying data (i.e., ontology visualisation); and the second category refers to meta-visualisations that summarise the ontology and data in order to provide users with generic insights about conceptual space and underlying data (i.e., tag-clouds, network visualisations etc.).

Finally, we would like to provide interested readers with the entry points to the work on different components of Optique, which all together form the OBDA framework for Big Data. The Optique framework includes components for *ontology and mapping management*, *time and streams*, *query transformation*, and *distributed query execution*. The ontology and mapping management component includes *bootstrapping* the ontology and mappings from existing schemas and models, mapping analysis and transformation, and supporting evolution and maintenance of the ontology and mappings (cf. [14]). The query transformation component is required to provide rewriting-based query answering over ontologies and optimisation with respect to the data source(s) (cf. [5]). The time and streams component is to support scalable temporal query answering, continuous query answering over streams of data, and stream-based event recognition (cf. [16]). Last but not the least, the distributed query execution component will provide distributed techniques for query planning and execution for one-time and continuous/streaming and temporal queries that scale to TBs of data (cf. [21]). The detailed description of the Optique, its architectural description, and the demos of aforementioned components can be found in [12, 19, 18].

7. ACKNOWLEDGMENTS

This research is funded by the Seventh Framework Program (FP7) of the European Commission under Grant Agreement 318338, "Optique". We would like to thank Martin G. Skjæveland for his help in preparing the dataset of NPD.

8. REFERENCES

[1] G. Barzdins, E. Liepins, M. Veilande, and M. Zviedris. Ontology Enabled Graphical Database Query Tool for End-Users. In *Eighth International Baltic Conference on Databases and Information Systems (DB&IS 2008)*, volume 187 of *Frontiers in Artificial Intelligence and Applications*, pages 105–116. IOS Press, 2009.

[2] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *3rd International Semantic Web User Interaction Workshop (SWUI 2006)*, 2006.

[3] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[4] P. Brusilovsky, A. Kobsa, and W. Nejdl, editors. *The adaptive web: methods and strategies of web personalization*. Springer-Verlag, Berlin, Heidelberg, 2007.

[5] D. Calvanese, I. Horrocks, E. Jiménez-Ruiz, E. Kharlamov, M. Meier, M. Rodriguez-Muro, and D. Zheleznyakov. On Rewriting and Answering Queries in OBDA Systems for Big Data (Short Paper). In *OWL Experiences and Directions Workshop (OWLED 2013)*, 2013.

[6] L. J. Campbell, T. A. Halpin, and H. A. Proper. Conceptual schemas with abstractions making flat conceptual schemas more comprehensible. *Data & Knowledge Engineering*, 20(1):39–85, 1996.

[7] T. Catarci. What happened when database researchers met usability. *Information Systems*, 25(3):177–212, 2000.

[8] T. Catarci, M. F. Costabile, S. Levialdi, and C. Batini. Visual query systems for databases: A survey. *Journal of Visual Languages and Computing*, 8(2):215–260, 1997.

[9] T. Catarci, P. Dongilli, T. Di Mascio, E. Franconi, G. Santucci, and S. Tessaris. An ontology based visual tool for query formulation support. In *16th European Conference on Artificial Intelligence (ECAI 2004)*, volume 110 of *Frontiers in Artificial Intelligence and Applications*, pages 308–312. IOS Press, 2004.

[10] J. Crompton. Keynote talk, the W3C Workshop on Semantic Web in Oil & Gas Industry: Houston, TX, USA, 9–10 December, 2008. available from http://www.w3.org/2008/12/ogws-slides/Crompton.pdf.

[11] R. G. Epstein. The TableTalk Query Language. *Journal of Visual Languages and Computing*, 2(2):115–141, 1991.

[12] M. Giese, D. Calvanese, I. Horrocks, Y. Ioannidis, H. Klappi, M. Koubarakis, M. Lenzerini, R. Moller, O. Ozcep, M. Rodriguez Muro, R. Rosati, R. Schlatte, A. Soylu, and A. Waaler. Scalable End-user Access to Big Data. In A. Rajendra, editor, *Big Data Computing*. Chapman and Hall/CRC, 2013.

[13] B. C. Grau, M. Giese, I. Horrocks, T. Hubauer, E. Jimenez-Ruiz, E. Kharlamov, M. Schmidt, A. Soylu, and D. Zheleznyakov. Towards Query Formulation and Query-Driven Ontology Extensions in OBDA Systems. In *OWL Experiences and Directions Workshop (OWLED 2013)*, 2013.

[14] P. Haase, I. Horrocks, D. Hovland, T. Hubauer, E. Jiménez-Ruiz, E. Kharlamov, J. Klüwer, C. Pinkel, R. Rosati, V. Santarelli, A. Soylu, and D. Zheleznyakov. Optique System: Towards Ontology and Mapping Management in OBDA Solutions. In *Workshop on Debugging Ontologies and Ontology Mappings (WoDOOM 2013)*, volume 999 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[15] A. Harth. VisiNav: A system for visual search and

navigation on web data. *Journal of Web Semantics*, 8(4):348–354, 2010.

[16] I. Horrocks, T. Hubauer, E. Jiménez-Ruiz, E. Kharlamov, M. Koubarakis, R. Möller, K. Bereta, C. Neuenstadt, O. Özçep, M. Roshchin, P. Smeros, and D. Zheleznyakov. Addressing Streaming and Historical Data in OBDA Systems: Optique's Approach (Statement of Interest). In *Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (Know@LOD 2013)*, volume 992 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[17] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods - A survey. *ACM Computing Surveys*, 39(4):10:1–10:43, 2007.

[18] E. Kharlamov, M. Giese, E. Jiménez-Ruiz, M. G. Skjæveland, A. Soylu, D. Zheleznyakov, T. Bagosi, M. Console, P. Haase, I. Horrocks, S. Marciuska, C. Pinkel, M. Rodriguez-Muro, M. Ruzzi, V. Santarelli, D. F. Savo, K. Sengupta, M. Schmidt, E. Thorstensen, J. Trame, and A. Waaler. Optique 1.0: Semantic Access to Big Data: The Case of Norwegian Petroleum Directorate's FactPages. In *International Semantic Web Conference (ISWC 2013) Posters and Demonstrations*, 2013.

[19] E. Kharlamov, E. Jiménez-Ruiz, D. Zheleznyakov, D. Bilidas, M. Giese, P. Haase, I. Horrocks, H. Kllapi, M. Koubarakis, O. Özcep, M. Rodríguez-Muro, R. Rosati, M. Schmidt, R. Schlatte, A. Soylu, and A. Waaler. Optique: Towards OBDA Systems for Industry. In *The Semantic Web: ESWC 2013 Satellite Events*, volume 7955 of *LNCS*. Springer, 2013.

[20] N. Khoussainova, Y. Kwon, W.-T. Liao, M. Balazinska, W. Gatterbauer, and D. Suciu. Session-based browsing for more effective query reuse. In *23rd international conference on Scientific and statistical database management (SSDBM 2011)*, volume 6809 of *LNCS*, pages 583–585. Springer-Verlag, 2011.

[21] H. Kllapi, D. Bilidas, I. Horrocks, Y. Ioannidis, E. Jiménez-Ruiz, E. Kharlamov, M. Koubarakis, and D. Zheleznyakov. Distributed Query Processing on the Cloud: the Optique Point of View (Short Paper). In *OWL Experiences and Directions Workshop (OWLED 2013)*, 2013.

[22] M. R. Kogalovsky. Ontology-Based Data Access Systems. *Programming and Computer Software*, 38(4):167–182, 2012.

[23] H. Lieberman, F. Paternò, and V. Wulf, editors. *End User Development*, volume 9 of *Human-Computer Interaction Series*. Springer, 2006.

[24] S. Madden. From Databases to Big Data. *IEEE Internet Computing*, 16(3):4–6, 2012.

[25] G. Marchionini. Exploratory search: From finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[26] K. Munir, M. Odeh, and R. McClatchey. Ontology-driven relational query formulation using the semantic and assertional capabilities of OWL-DL. *Knowledge-based Systems*, 35:144–159, 2012.

[27] J. F. Nunameker, R. O. Briggs, and G.-J. de Vreede. From Information Technology to Value Creation Technology. In *Information Technology and the Future Enterprise: New Models for Managers*, pages 102–124. Prentice-Hall, New York, 2001.

[28] M. Rodriguez-Muro and D. Calvanese. High Performance Query Answering over DL-Lite Ontologies. In *Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 308–318. AAAI Press, 2012.

[29] M. Rodriguez-Muro and D. Calvanese. Quest, a System for Ontology Based Data Access. In *OWL Experiences and Directions Workshop 2012 (OWLED 2012)*, volume 849 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[30] F. Ruiz and J. R. Hilera. Using Ontologies in Software Engineering and Technology. In C. Calero, F. Ruiz, and M. Piattini, editors, *Ontologies for Software Engineering and Software Technology*, pages 49–102. Springer-Verlag, 2006.

[31] M. C. Schraefel, M. Wilson, A. Russell, and D. A. Smith. mSpace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM*, 49(4):47–49, 2006.

[32] B. Shneiderman. Direct Manipulation: A Step Beyond Programming Languages. *Computer*, 16(8):57–69, 1983.

[33] M. G. Skjæveland, E. H. Lian, and I. Horrocks. Publishing the Norwegian Petroleum Directorate's FactPages as Semantic Web Data. In *The Semantic Web – ISWC 2013*, volume 8219 of *LNCS*, 2013.

[34] A. Soylu, F. Moedritscher, F. Wild, P. De Causmaecker, and P. Desmet. Mashups by orchestration and widget-based personal environments: Key challenges, solution strategies, and an application. *Program: Electronic Library And Information Systems*, 46(4):383–428, 2012.

[35] B. Suh and B. B. Bederson. OZONE: A Zoomable Interface for Navigating Ontology Information. In *Working Conference on Advanced Visual Interfaces (AVI 2002)*, pages 139–143. ACM, 2002.

[36] A. H. M. Ter Hofstede, H. A. Proper, and T. P. Van Der Weide. Query formulation as an information retrieval problem. *Computer Journal*, 39(4):255–274, 1996.

[37] D. Tunkelang and G. Marchionini. *Faceted Search*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan and Claypool Publishers, 2009.

[38] R. W. White, B. Kules, S. M. Drucker, and M. C. Schraefel. Supporting exploratory search. *Communications of the ACM*, 49(4):37–39, 2006.

[39] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *SIGCHI Conference on Human Factors in Computing Systems (CHI 2003)*, pages 401–408. ACM, 2003.

[40] C. Yu and H. V. Jagadish. Schema summarization. In *32nd international conference on Very large data bases (VLDB'06)*, pages 319–330. VLDB Endowment, 2006.

[41] K. Zheng, Q. Mei, and D. A. Hanauer. Collaborative search in electronic health records. *Journal of the American Medical Informatics Association*, 18(3):282–291, 2011.